

IN THE  
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Joan Raventos

Confirmation No.: 2088

Application No.: 09/872,442

Examiner: N. R. Shah

Filing Date: 06/01/2001

Group Art Unit: 2195

Title: SYSTEM AND METHOD FOR ENABLING TRANSACTION-BASED SERVICE UTILIZING  
NON-TRANSACTIONAL RESOURCES

Mail Stop Appeal Brief-Patents  
Commissioner For Patents  
PO Box 1450  
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 07/06/2005.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( ) one month	\$120.00
( ) two months	\$450.00
( ) three months	\$1020.00
( ) four months	\$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of \$500.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the U.S. Postal Service as Express Mail, Airbill No. EV629198456US, in an envelope addressed to: MS Appeal Briefs - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.  
Date of Deposit: September 6, 2005

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number \_\_\_\_\_ on \_\_\_\_\_

Number of pages:

Typed Name: **Rhyllis Ewing**

Signature: 

Respectfully submitted,

**Joan Raventos**

By 

**Jody C. Bishop**

Attorney/Agent for Applicant(s)

Reg. No. **44,034**

Date: **09/06/2005**

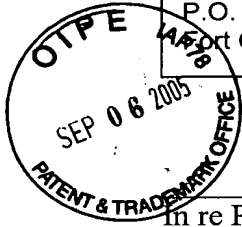
Telephone No.: **(214) 855-8007**

09-07-05

AF-  
Tfw

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, Colorado 80527-2400

Docket No.: 10012815-1  
(PATENT)



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of:  
Joan S. Raventos

Application No.: 09/872,442

Confirmation No.: 2088

Filed: June 1, 2001

Art Unit: 2195

For: SYSTEM AND METHOD FOR ENABLING  
TRANSACTION-BASED SERVICE  
UTILIZING NON-TRANSACTIONAL  
RESOURCES

Examiner: N. R. Shah

**APPEAL BRIEF**

MS Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on July 6, 2005, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

- |       |   |
|-------|---|
| I.    | Real Party In Interest                        |
| II.   | Related Appeals and Interferences             |
| III.  | Status of Claims                              |
| IV.   | Status of Amendments                          |
| V.    | Summary of Claimed Subject Matter             |
| VI.   | Grounds of Rejection to be Reviewed on Appeal |
| VII.  | Argument                                      |
| VIII. | Claims  |

09/08/2005 TBESHAH1 00000020 082025 09872442

01 FC:1402 500.00 DA  
25557203.1

IX.	Evidence
X.	Related Proceedings
Appendix A	Claims

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 28 claims pending in application.

B. Current Status of Claims

1. Claims pending: 1, 2, 4-24 and 27-31
2. Claims canceled: 3, 25 and 26
3. Claims withdrawn from consideration but not canceled: None
4. Claims allowed: None
5. Claims rejected: 1, 2, 4-24 and 27-31
6. Claims objected to: None

C. Claims On Appeal

The claims on appeal are claims 1, 2, 4-24 and 27-31

#### IV. STATUS OF AMENDMENTS

Appellant filed a Response to Final Office Action on April 27, 2005 that did not make any amendments. The amendments made in an Amendment in Response to Non-Final Office Action filed on November 12, 2004 are reflected in the listing of the claims given in Appendix A.

#### V. SUMMARY OF CLAIMED SUBJECT MATTER

According to one claimed embodiment of the present invention, a system for performing a desired functional service (102-110 of Fig. 1; page 9, lines 23-26) as a transaction (page 10, lines 14-15 and 23-26) utilizing one or more non-transactional resources (214 of Fig. 2; page 17 lines 10-18; page 13, lines 6-10) comprises: one or more non-transactional resources; at least one component (211 of Fig. 2, page 16, lines 16-18) that defines one or more tasks executable by at least one of said one or more non-transactional resources; and resource manager (213 of Fig. 2) operable to control execution of said one or more tasks defined by said at least one component as a transaction for activation of a service (page 17, lines 26-28).

According to another claimed embodiment, a method of performing a functional service (102-110 of Fig. 1; page 9, lines 23-26) as a transaction utilizing one or more non-transactional resources (214 of Fig. 2; page 17 lines 10-18; page 13, lines 6-10) within a computing environment (200 of Fig.2), comprises the steps of: at least one component (211 of Fig. 2, page 16, lines 16-18) defining one or more tasks executable by at least one of said one or more non-transactional resources; receiving at a resource manager (213 of Fig. 2) a request for performance of a plurality of tasks as a transaction (page 17, lines 23-26); and said resource manager controlling execution (page 17, line 26 through page 18, line 3) of said at least one component to perform said plurality of tasks as a transaction for service provisioning.

According to another claimed embodiment, a resource manager (213 of Fig. 2) operable to control execution (page 17, line 26 through page 18, line 3) of tasks by one or more non-transactional resources (214 of Fig. 2; page 17 lines 10-18; page 13, lines 6-10) to perform said tasks as a transaction, comprises: code for receiving a request (page 19, line 21 through page 20, line 2) for performance of a plurality of tasks; code for controlling (page 17,

line 26 through page 18, line 3) one or more non-transactional resources to perform said plurality of tasks as a transaction, wherein said code for controlling one or more non-transactional resources includes code (page 21, line 26 through page 22, line 3) for invoking performance of a task by said one or more non-transactional resources, and wherein said code for invoking performance of a task includes code for calling a function defined by a plugin component (325 of Fig. 3B; page 20, lines 19-22) that is communicatively coupled to said one or more non-transactional resources.

According to another claimed embodiment of the invention, a system comprises: non-transactional resources (214 of Fig. 2; page 17 lines 10-18; page 13, lines 6-10) for providing a plurality of different services; a plugin (325 of Fig. 3B; page 20, lines 19-22) for each of said plurality of different services that defines one or more tasks for activating the respective service; and resource manager (213 of Fig. 2; page 17, line 26 through page 18, line 3) operable to control execution of said plugins to selectively activate multiple ones of the plurality of different services as a transaction.

According to another claimed embodiment of the invention, a system comprises: at least one non-transactional resource (214 of Fig. 2; page 17 lines 10-18; page 13, lines 6-10) for providing at least one service associated with a web hosting service (page 19, lines 12-13); a plugin (325 of Fig. 3B; page 20, lines 19-22) for said at least one non-transactional resource that defines one or more tasks for activating the at least one associated service for a given web hosting service; and resource manager (213 of Fig. 2; page 17, line 26 through page 18, line 3) operable, when activating said given web hosting service, to control execution (page 17, line 26 through page 18, line 3) of said plugin to activate said at least one associated service as a transaction with activating said given web hosting service.

According to another claimed embodiment of the invention, a service that is activated includes a plurality of different services associated with a web hosting service (page 9, lines 4-6). According to other claimed embodiments of the invention, transactions and services include: web hosting service, ftp service, database service, software application service, Domain Name Service (DNS), directory service, monitoring service, managing service, monitoring Quality of Service (QoS), usage measurement service, billing service, software application service and LDAP service (page 9, lines 12-23).

According to another claimed embodiment of the invention, non transactional resources are resources of an Internet Data Center (IDC) (page 9, lines 4-6). According to other claimed embodiments of the invention transactional protocol is X/Open XA protocol (page 11, lines 15-17). According to other claimed embodiments of the invention, at least one component of the system is a plugin (page 13, lines 20-23).

## VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 2, 4-24 and 27-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,872,971 to Knapman et al. (“Knapman”) in view U.S. Patent 6,526,416 to Long (“Long”).

## VII. ARGUMENT

Below, Appellant argues many of the non-allowed claims separately. Thus, Appellant respectfully asserts that separately argued claims do not stand or fall together, *see* 37 C.F.R. § 41.37(c)(1)(vii)

### I. Rejections under 35 U.S.C. § 103(a)

To establish a *prima facie* case of obviousness, three basic criteria must be met. *See* M.P.E.P. § 2143. First, there must be some suggestion or motivation, either in the applied references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the applied references must teach or suggest all the claim limitations. Without conceding any other criteria, Appellant respectfully asserts that the applied combination of *Knapman* and *Long* does not teach or suggest all the claim limitations.

### A. Independent Claim 1 and Dependent Claims 4, 6, and 9-14

Independent claim 1 recites, in part, “one or more non-transactional resources; at least one component that defines one or more tasks executable by at least one of said one or more non-transactional resources”. The Final Office Action and the Advisory Action assert that *Knapman* teaches these elements of claim 1, *see* item 4 on page 2 and item 36 on page 8 of

the current Office Action and the continuation sheet of the Advisory Action. Appellant respectfully disagrees, as discussed below.

As discussed in the Amendment of November 12, 2004, *Knapman* fails to address non-transactional resources. Rather, *Knapman* teaches a system that enables “operation requests from an application program which implements one transactional model to be translated to operations which are meaningful to a transaction processing manager which implements a different transactional model.” Col. 5, lines 29-34. *Knapman* recognizes that different transactional models exist, such as the CICS transactional model and the OTS transactional model, *see* Col. 3, lines 13-19. The operational mismatch between the different transactional models “currently prevents interoperation between data processing resources which implement the different models.” Col. 3, lines 20-22.

Thus, *Knapman* teaches a system that uses transactional resources and addresses compatibility of different transactional models rather than teaching a system that uses non-transactional resources as recited by claim 1.

In response to this argument, the Final Office Action asserts that *Knapman* “teaches both transactional and nontransactional information and how each can be distinctly marked (col. 2 lines 51-55 col. 9 lines 40-42)”, *see* item 36 on page 8 of Final Office Action. The cited portions of *Knapman* describe an OTS transactional system, which provides an unchained model, and a CICS transactional system that implements a chained model. For instance, column 2, lines 26-55 of *Knapman* provides:

A number of different transaction processing models are known in the art. Some transaction processing systems require explicit transactional demarcation--i.e. an application program must issue the relevant API commands for operations to begin, commit or rollback a transaction. This requirement is a feature of the Object Management Group, Inc.'s (OMG's) Object Transaction Service (OTS) specification--a specification for a proposed service which supports transactional behaviour in a distributed heterogeneous environment based on the OMG Common Object Request Broker Architecture (CORBA). CORBA defines the interactions between objects of an object oriented system (see below), and in particular between distributed objects in different systems. The OTS exploits object oriented programming to encapsulate the processing performed under transactional scope, allowing a programmer (or similar person) to designate certain classes of operations as transactional. The unchained model of transactions is supported since a client

program can explicitly start, suspend and resume transactions. Invoking a transactional operation outside a transaction causes the operation to be performed outside the scope of any transaction. In the unchained model, a program can explicitly enter and leave the scope of transactions so that some of its execution might be within the scope of one transaction, some within the scope of another, and some outside the scope of any transaction. Unlike the non-object-oriented unchained model, in which a program can freely mix transactional and non-transactional work, OTS requires that the transactional operations be distinctly marked.

Thus, this portion of *Knapman* describes that different transactional processing models are known. It further describes that OTS is one transactional processing system which provides an unchained model in which a program can explicitly enter and leave the scope of transactions so that some of its execution might be within the scope of one transaction, some within the scope of another, and some outside the scope of any transaction. To support such mixing of transactional and non-transactional work, OTS requires that the transactional operations be distinctly marked within a program.

Thus, OTS provides a transactional processing system. While operations of a program may be performed outside the scope of a transaction, OTS provides transactional resources that are capable of performing transactional tasks when so desired. That is, if a transaction is desired, the transactional resources of OTS are used for achieving such transaction. Thus, OTS provides transactional resources that can be used when tasks are desired to be performed as a transaction.

At column 2, line 60 through column 3, line 7, *Knapman* provides:

The CICS on-line transaction processing programs, which are commercially available from International Business Machines Corporation, support "chained" transactions across cooperating systems. Chained transactions have the property that a program (e.g. a CICS application) is always within the scope of some transaction, and a system that implements the chained model can maintain the rule that all transactional operations must be executed under its control (by means of transaction programs in the case of CICS systems). A CICS transaction begun in one system may include operations performed by other systems, with demarcation of the transaction (e.g. commit processing) being controlled by a CICS system without requiring application-initiated demarcation operations.

Thus, CICS is a transactional processing system that supports chained transactions wherein a program is always within the scope of some transaction, as opposed to OTS where



the program can enter and leave transactions. Thus, CICS also provides transactional resources. *Knapman* further describes at column 3, lines 13-22:

It is an important distinction between the CICS transactional model and the OTS transactional model that CICS servers manage units of work and support chained transactions whereas OTS requires client application control and implements an unchained transaction model, since the respective transactions look very different, and require different action, from the client application. This mismatch between the transactional models currently prevents interoperation between data processing resources which implement the different models. (Emphasis added).

Thus, *Knapman* is concerned with the mismatch between the CICS and OTS transactional models. For instance, at column 8, lines 49-55 *Knapman* states:

While the invention relates generally to achieving interoperability between data processing resources which implement different transactional models, the invention is particularly useful in providing a solution to the mismatch between resources which support chained transactions, such as CICS transaction processing systems, and the unchained transactional model implemented in OTS. (Emphasis added).

*Knapman* describes operation of CICS transactions as follows:

CICS transactions are generally begun by an end user signing on to a CICS system and then invoking a particular application that they intend to use, generally by typing the transaction identification code (id) or using a predefined program function key. CICS looks up the transaction identifier in its internal Program Control Table where it finds out which program to invoke first to execute the requested transaction. Subsequent delimiting of the transaction is controlled by the CICS system's implicit management of units of work; CICS coordinates updates when a syncpoint request is received.

Client-server programming in CICS uses two application programming interfaces (APIs) that provide external access to CICS facilities, allowing non-CICS applications to gain access to CICS facilities and data:

External call interface (ECI)

External presentation interface (EPI)

FIG. 2 illustrates the prior art use of the external interfaces 100 by a non-CICS application 110 which is located in a client system 120 and is using the facilities of CICS 130 in a server system 140. The CICS client software 150 processes the application's EPI and ECI requests, and transmits them to the server system using an appropriate communication protocol. Although FIG. 2 shows separate client and server systems, the configuration may be on a single

workstation. The installation of a CICS client program on a data processing system enables that system to connect to an appropriate CICS server program. (Col. 8, line 65 – Col. 9, line 25)...

The ECI allows a non-CICS application to call a CICS program in a CICS server and to initiate a CICS transaction. The application can be connected to several servers at the same time, and it can have several program calls outstanding at the same time. The CICS program can access and update all CICS resources except performing terminal input/output. (Col. 9, lines 41-46)...

As compared with the basic model of chained transactions (where a program is always within the scope of a transaction), the CICS ECI affords greater flexibility by extending the concept of a chained transaction with the concept of a "mirror transaction". The CICS ECI allows a client program to execute outside the scope of a transaction but to invoke several CICS application programs under the scope of one or more transactions. The client program identifies these transactions by means of a logical unit of work identifier. An invocation of an application program (when the task implemented by that application program is selected to be performed) is termed a transaction scheduling. Much of the simplicity of the chained transactional model is retained.

The transaction programs, however, must always be separate from the client program, these being executed under the control of a server program. The transactional programs are always within the scope of a transaction as a consequence of the chained transaction model. (Emphasis added). (Col. 9, line 64 – col. 10, line 14).

In view of the above, *Knapman* does not address using non-transactional resources for performing a transaction. Rather, *Knapman* addresses use of transactional resources, which may be invoked by an application program to perform a desired transaction. For instance, a non-CICS application may interface with a CICS application to invoke a transaction to be performed via the CICS system's transactional resources. If a transaction is desired in *Knapman*, transactional resources are used by invoking a transactional processing system, such as CICS, rather than using non-transactional resources as recited by claim 1.

*Knapman* further addresses a technique for translating from a first transactional model to a different transactional model to enable interoperation of transactional resources of each model (e.g., OTS transactional resources and CICS transactional resources), but *Knapman* fails to teach or suggest non-transactional resources in the manner recited in claim 1.

Thus, because the current Office Action relies on *Knapman* as teaching or suggesting the above elements of claim 1, the present rejection is improper as the Office Action has failed to establish a prima facie case of obviousness.

Although the Office Actions have relied upon *Long* as teaching this element, *Long* appears to address utilizing non-transactional resources in performing transactional operations, *see e.g.*, Col. 1, lines 5-10; col. 2, lines 60-62; col. 3, lines 2-6; and col. 10, lines 55-61. However, claim 1 recites “resource manager operable to control execution of said one or more tasks defined by said at least one component as a transaction for activation of a service” (emphasis added). *Long* fails to teach or suggest controlling execution of defined tasks by non-transactional resources for activating a service.

In response to this argument raised in the amendment of November 12, 2004, the Final Office Action asserts that *Long* “teaches a resource manager able to activate a transaction for service (col. 10 lines 35-44; col. 27 lines 4-27)”, *see* item 36 on page 9 of Final Office Action. While *Long* proposes a resource manager that supports performing transactions, *Long* does not teach or suggest activating a service as a transaction. The Final Office Action notes that *Long*’s resource manager is able to “activate a transaction”. However, this is not what claim 1 recites. Rather, claim 1 recites “resource manager operable to control execution of said one or more tasks defined by said at least one component as a transaction for activation of a service” (emphasis added). Again, while the resource manager of *Long* supports transactions, it does not teach or suggest performing one or more tasks as a transaction for activation of a service, such as activation of a web hosting service for hosting a new web site for a customer (*see e.g.*, page 2, line 1 – page 3, line 7 of the present application).

The Advisory Action further asserts that *Long* “clearly teaches the use of a service (abstract) and a activation process (col. 12 lines 34-55; col. 1 lines 20-27).” Appellant notes that the services listed in the section of *Long* cited by the Advisory Action, such as class registrations, travel reservations, services at a bank, and the others, are services that are already in place and are merely being accessed – they are not services being activated. That is, while *Long* teaches that these services may be used as a transaction, *Long* does not teach or suggest activating any of these services.

*Long* presumes that all services are already available, that is that they are already activated, and merely addresses how the services may be used (e.g. for making travel reservations, etc.). Appellant notes that the mere use of a service is not the same as the activation of a service.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 1. Therefore claim 1 is not obvious under 35 U.S.C. § 103(a) over this combination. Dependent claims 4, 6 and 9-14 depend from claim 1 and thus inherit all the limitations of claim 1. Appellant respectfully submits that claims 1, 4, 6 and 9-14 are patentable over the 35 U.S.C. § 103(a) rejection of record.

#### **B. Independent Claim 15 and Dependent Claims 16-20**

Independent claim 15 recites, in part, “at least one component defining one or more tasks executable by at least one of said one or more non-transactional resources”. The current Office Action rejects claim 15 “based on the same rejection for claim 1”. Thus, as discussed above with claim 1, the Office Action relies on *Knapman* as teaching the above element. As further discussed above with claim 1, *Knapman* fails to teach or suggest one or more non-transactional resources, and thus also fails to teach or suggest at least one component defining one or more tasks executable by at least one of the non-transactional resource(s).

Thus, because the current Office Action relies on *Knapman* as teaching or suggesting the above element of claim 15, the present rejection is improper as the Office Action has failed to establish a prima facie case of obviousness.

As discussed above with claim 1, *Long* appears to address utilizing non-transactional resources in performing transactional operations, *see e.g.*, Col. 1, lines 5-10; col. 2, lines 60-62; col. 3, lines 2-6; and col. 10, lines 55-61, *Long* fails to teach or suggest “said resource manager controlling execution of said at least one component to perform said plurality of tasks as a transaction for service provisioning” (emphasis added), as recited by claim 15. While *Long* teaches a resource manager, *Long* does not teach or suggest controlling execution of a plurality of tasks by a non-transactional resource as a transaction “for service provisioning”. Appellant notes that the mere use of a service is not “service provisioning”.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 15. Therefore, claim 15 is not obvious under 35 U.S.C. § 103(a) over this combination. Dependent claims 16-20 depend from claim 15 and thus inherits all the limitations of claim 15. Appellant respectfully submits that claims 15 and 16-20 are patentable over the 35 U.S.C. § 103(a) rejection of record.

### **C. Independent Claim 23 and Dependent Claim 24**

Independent claim 23 recites, in part, “code for controlling one or more non-transactional resources to perform said plurality of tasks as a transaction, wherein said code for controlling one or more non-transactional resources includes code for invoking performance of a task by said one or more non-transactional resources, and wherein said code for invoking performance of a task includes code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources” (emphasis added). The combination of *Knapman* and *Long* fails to teach or suggest at least the above element of claim 23. As described with claim 1 above, *Knapman* does not address non-transactional resources. While *Long* addresses non-transactional resources, *Long* fails to teach or suggest calling a function defined by a plugin component that is communicatively coupled to the non-transactional resources for invoking performance of a task by the non-transactional resources. No such plugin component is taught or suggested by *Long*.

The Final Office Action asserts that *Knapman* teaches “wherein said code for invoking performance of a task includes code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources” citing to col. 10, lines 1-5, col. 5, lines 19-27, and col. 6, lines 58-62 of *Kapman*, see item 28 on page 7 of Final Office Action. Col. 10, lines 1-5 of *Knapman* provides:

The CICS ECI allows a client program to execute outside the scope of a transaction but to invoke several CICS application programs under the scope of one or more transactions. The client program identifies these transactions by means of a logical unit of work identifier.

This does not teach or suggest code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources.

Rather, the ECI is an interface for the CICS transactional processing system, which provides transactional resources, rather than non-transactional resources.

Col. 5, lines 10-27 of *Knapman* provides:

According to a first aspect of the present invention, there is provided a data processing system including data processing resources which implement a first transactional model, in which all transactional operations are performed within transactional scope, and means for interfacing between said resources and data processing resources which implement a second transactional model, in which transactional operations may be invoked and performed outside of transactional scope, thereby to enable interoperation between said different resources in the processing of a transaction, the means for interfacing including: means for mapping between transactional operations according to the second model and transactional operations according to the first model, said mapping including designating transactional operations of said second model which are invoked outside of a first transaction as separate transactions according to the first model which are isolated from said first transaction; and means for maintaining a list of said operation mappings for the transaction.

This does not teach or suggest code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources. Rather, this proposes an interface between a first transactional model and a second transactional model. Such transactional models each use transactional resources rather than non-transactional resources.

Col. 6, lines 56-65 of *Kapman* provides:

The external interface of the transaction processing program may be an external interface of a client program of a client-server transaction service. The isolated transactions will possess all of the properties required of a non-transactional operation, whereas the additional properties that they can be rolled back or committed may be invisible and unavailable to the client program. However, the client can rely on marking and encapsulation according to the second model to facilitate identification of the transactional operations and their performance by a server program.

This also does not teach or suggest code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources. No plugin that is coupled to a non-transactional resource is mentioned. Further, this proposes an external interface of a client program of a transaction service that enables interoperation between two different transactional models. For instance, it allows for marking to identify

transactional operations to be performed by transactional resources of a transactional model. Thus, if an operation is identified as transactional, the transactional resources of a transactional model (e.g., the CICS system or OTS system) can be used for processing such transaction.

Again, *Knapman* does not teach or suggest using non-transactional resources for performing a transaction, and fails to teach or suggest code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 23. Therefore, claim 23 is not obvious under 35 U.S.C. § 103(a) over this combination. Dependent claim 24 depend from claim 23 and thus inherits all the limitations of claim 23. Appellant respectfully submits that claims 23 and 24 are patentable over the 35 U.S.C. § 103(a) rejection of record.

#### **D. Independent Claim 27**

Claim 27 recites:

A system comprising:  
non-transactional resources for providing a plurality of different services;  
a plugin for each of said plurality of different services that defines one or more tasks for activating the respective service; and  
resource manager operable to control execution of said plugins to selectively activate multiple ones of the plurality of different services as a transaction. (Emphasis added).

The combination of *Knapman* and *Long* fails to teach or suggest each of the above elements of claim 27. For instance, *Knapman* does not teach or suggest non-transactional resources for providing a plurality of different services and using a resource manager to control execution of plugins to selectively activate multiple ones of the plurality of different services as a transaction. Rather, if an operation is desired to be performed as a transaction in *Knapman*, transactional resources of a transactional model are used for performing such transaction. *Knapman* focuses on enabling use of such transactional resources of a plurality of different types of transactional models, such as OTS and CICS. *Knapman* does not teach

or suggest performing any operations, including those for activating services, as a transaction using non-transactional resources.

As described above, *Long* appears to propose use of non-transactional resources. However, *Long* fails to teach or suggest a “resource manager operable to control execution of said plugins to selectively activate multiple ones of the plurality of different services as a transaction” (emphasis added). *Long* does not teach or suggest activating different services as a transaction, such as activating ones of “web hosting service, database service, software application service, DNS service, LDAP service, monitoring service, management service, quality of service (QoS) service, usage measurement service, and billing service”, as further recited in claim 28. While *Long* mentions services that may be used as a transaction, such as for making travel reservations, *Long* does not address activation of such services as a transaction.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 27, and thus claim 27 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **E. Independent Claim 29**

Claim 29 recites:

A system comprising:  
at least one non-transactional resource for providing at least one service associated with a web hosting service;  
a plugin for said at least one non-transactional resource that defines one or more tasks for activating the at least one associated service for a given web hosting service; and  
resource manager operable, when activating said given web hosting service, to control execution of said plugin to activate said at least one associated service as a transaction with activating said given web hosting service. (Emphasis added).

The combination of *Knapman* and *Long* fails to teach or suggest each of the above elements of claim 29. For instance, *Knapman* does not teach or suggest at least one non-transactional resources for providing at least one service associated with a web hosting service and using a resource manager to control execution of plugins to activate the at least one associated service as a transaction when activating a given web hosting service. Rather,



if an operation is desired to be performed as a transaction in *Knapman*, transactional resources of a transactional model are used for performing such transaction. *Knapman* focuses on enabling use of such transactional resources of a plurality of different types of transactional models, such as OTS and CICS. *Knapman* does not teach or suggest performing any operations, including those for activating services associated with a web hosting service, as a transaction using non-transactional resources.

As described above, *Long* appears to propose use of non-transactional resources. However, *Long* fails to teach or suggest a “resource manager operable, when activating said given web hosting service, to control execution of said plugin to activate said at least one associated service as a transaction with activating said given web hosting service.” *Long* does not teach or suggest activating at least one associated service as a transaction when activating a web hosting service, such as activating ones of “database service, software application service, DNS service, LDAP service, monitoring service, management service, quality of service (QoS) service, usage measurement service, and billing service”, as further recited in claim 31.

Indeed, *Long* does not address activation of a web hosting service at all, and certainly fails to teach or suggest activating at least one associated service as a transaction with activating a web hosting service.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 29, and thus claim 29 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **F. Claim 2**

Claim 2 depends from claim 1, and is thus allowable for, at least, the reasons set forth with respect to claim 1. Further, claim 2 presents additional limitations. Claim 2 recites:

The system of claim 1 wherein said transaction includes performance of at least one of the services selected from the group consisting of: web hosting service, ftp service, database service, software application service, Domain Name Service (DNS), directory service, monitoring service, managing service, monitoring Quality of Service (QoS), usage measurement service, and billing service.

Neither *Long* nor *Knapman* teaches or suggests activating a web hosting service, or activating any of the other services listed in claim 2.

The Final Office Action asserts that *Long* meets the elements of claim 2, and cites to col. 7, lines 17-35, along with Fig. 1. However, the cited sections of *Long* merely indicate that internet is available to connect computers. Thus, *Long* does not teach or suggest the further elements of claim 2. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 2.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 2, and thus claim 2 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **G. Claim 5**

Claim 5 depends from claim 1, and is thus allowable for, at least, the reasons set forth with respect to claim 1. Further, claim 5 presents additional limitations. Claim 5 recites “said non-transactional resources are resources of an Internet Data Center (IDC).” Neither *Long* nor *Knapman* teaches or suggests that non-transactional resources are resources of an IDC. The Final Office Action asserts that *Long* meets the elements of the claims, and cites to col. 7, lines 17-35, along with Fig. 1. However, the cited sections of *Long* does not teach or suggest an IDC as required by claim 5. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 5.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 5, and thus claim 5 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **H. Claim 7**

Claim 7 depends from claim 1, and is thus allowable for, at least, the reasons set forth with respect to claim 1. Further, claim 7 presents additional limitations. Claim 7 recites “said transactional protocol is X/Open XA protocol” Neither *Long* nor *Knapman* teaches or suggests that the transactional protocol is X/Open XA protocol. The Final Office Action asserts that *Long* meets the elements of the claims, and cites to col. 8, line 65 to col 9, line 13.

However, the cited section of *Long* does not teach or suggest a X/Open XA protocol as required by claim 7. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 7.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 7, and thus claim 7 is not obvious under 35 U.S.C. § 103(a) over this combination.

### **I. Claim 8**

Claim 8 depends from claim 1, and is thus allowable for, at least, the reasons set forth with respect to claim 1. Further, claim 8 presents additional limitations. Claim 8 recites “at least one component is a plugin.” Neither *Long* nor *Knapman* teaches or suggests that at least one component of the system is a plugin as required by claim 8.

The Final Office Action asserts that *Long* meets the elements of claim 8, and cites to col. 8, line 65 through col. 9, line 13. However, the cited section of *Long* merely states that “By convention, the interfaces of a COM object are illustrated graphically as a plug-in jack,” *see Long*, col. 9, lines 7-9. Thus, *Long* does not teach or suggest the further elements of claim 8. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 8.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 8, and thus claim 8 is not obvious under 35 U.S.C. § 103(a) over this combination.

### **J. Claim 21**

Claim 21 depends from claim 15, and is thus allowable for, at least, the reasons set forth with respect to claim 15. Further, claim 21 presents additional limitations. Claim 21 recites “said transactional protocol is X/Open XA protocol.” Neither *Long* nor *Knapman* teaches or suggests that the transactional protocol is X/Open XA protocol. The Final Office Action asserts that *Long* meets the elements of the claims, and cites to col. 8, line 65 to col 9, line 13. However, the cited section of *Long* does not teach or suggest a X/Open XA protocol

as required by claim 21. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 21.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 21, and thus claim 21 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **K. Claim 22**

Claim 22 depends from claim 15, and is thus allowable for, at least, the reasons set forth with respect to claim 15. Further, claim 22 presents additional limitations. Claim 22 recites “said at least one component is a plugin component.” Neither *Long* nor *Knapman* teaches or suggests that at least one component of the system is a plugin as required by claim 22.

The Final Office Action asserts that *Long* meets the elements of claim 22, and cites to col. 8, line 65 through col. 9, line 13. However, the cited section of *Long* merely states that “By convention, the interfaces of a COM object are illustrated graphically as a plug-in jack,” *see Long*, col. 9, lines 7-9. Thus, *Long* does not teach or suggest the further elements of claim 22. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 22.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 22, and thus claim 22 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **L. Claim 28**

Claim 28 depends from claim 27, and is thus allowable for, at least, the reasons set forth with respect to claim 27. Further, claim 28 presents additional limitations. Claim 28 recites:

The system of claim 27 wherein said plurality of different services comprise ones selected from the group consisting of:  
wherein said plurality of different services comprise ones selected from the group consisting of: web hosting service, database service, software application service, DNS service, LDAP service, monitoring service,

management service, quality of service (QoS) service, usage measurement service, and billing service.

Neither *Long* nor *Knapman* teaches or suggests activating a web hosting service, or activating any of the other services listed in claim 28.

The Final Office Action asserts that *Long* meets the elements of the claim 28, and cites to col. 7, lines 17-35, along with Fig. 1. However, the cited sections of *Long* merely indicate that internet is available to connect computers. Thus, *Long* does not teach or suggest the further elements of claim 28. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 28.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 28, and thus claim 28 is not obvious under 35 U.S.C. § 103(a) over this combination.

#### **M. Claim 30**

Claim 30 depends from claim 29, and is thus allowable for, at least, the reasons set forth with respect to claim 29. Further, claim 30 presents additional limitations. Claim 29 recites “said at least one associated service includes a plurality of different services associated with a web hosting service.” Thus, *Long* does not teach or suggest the further elements of claim 30. Neither *Long* nor *Knapman* teaches or suggests activating a web hosting service, or activating any of the other services listed in claim 30.

The Final Office Action asserts that *Long* meets the elements of the claim 28, and cites to col. 7, lines 17-35, along with Fig. 1. However, the cited sections of *Long* merely indicate that internet is available to connect computers. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 30.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 30, and thus claim 30 is not obvious under 35 U.S.C. § 103(a) over this combination.

**N. Claim 31**

Claim 31 depends from claim 30, and is thus allowable for, at least, the reasons set forth with respect to claim 30. Further, claim 31 presents additional limitations. Claim 31 recites:

The system of claim 30 wherein the plurality of different associated services comprise ones selected from the group consisting of:  
database service, software application service, DNS service, LDAP service, monitoring service, management service, quality of service (QoS) service, usage measurement service, and billing service.

Neither *Long* nor *Knapman* teaches or suggests activating any of the other services listed in claim 31.

The Final Office Action asserts that *Long* meets the elements of the claim 28, and cites to col. 7, lines 17-35, along with Fig. 1. However, the cited sections of *Long* merely indicate that internet is available to connect computers. Thus, *Long* does not teach or suggest the further elements of claim 31. *Knapman* is not relied upon by the Office Action as meeting the limitations of claim 31.

In view of the above, the applied combination of *Knapman* and *Long* fails to teach or suggest all elements of claim 31, and thus claim 31 is not obvious under 35 U.S.C. § 103(a) over this combination.

**VIII. CLAIMS**

A copy of the claims involved in the present appeal is attached hereto as Appendix A. As indicated above, the claims in Appendix A reflect the most recent amendment filed by Appellant on November 12, 2004, as well as the most recent response filed on April 27, 2005.

**IX. EVIDENCE**

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the Examiner is being submitted.

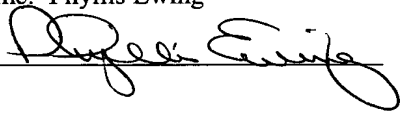
X. RELATED PROCEEDINGS

No related proceedings are referenced in II. Above. Therefore, copies of decisions in related proceedings are not provided.

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV629198456US in an envelope addressed to: M/S Appeal Brief, Director for Patents, Alexandria, VA 22313.

Date of Deposit: 09/06/2005

Typed Name: Phyllis Ewing

Signature: 

Respectfully submitted,

By: 

Jody C. Bishop

Attorney/Agent for Appellant(s)

Reg. No. 44,034

Date: 09/06/2005

Telephone No. (214) 855-8007

**APPENDIX A**

**Claims Involved in the Appeal of Application Serial No. 09/872,442**

1. A system for performing a desired functional service as a transaction utilizing one or more non-transactional resources, said system comprising:  
one or more non-transactional resources;  
at least one component that defines one or more tasks executable by at least one of said one or more non-transactional resources; and  
resource manager operable to control execution of said one or more tasks defined by said at least one component as a transaction for activation of a service.
2. The system of claim 1 wherein said transaction includes performance of at least one of the services selected from the group consisting of: web hosting service, ftp service, database service, software application service, Domain Name Service (DNS), directory service, monitoring service, managing service, monitoring Quality of Service (QoS), usage measurement service, and billing service.
3. (Canceled)
4. The system of claim 1 comprising:  
a plurality of said non-transactional resources, wherein said non-transactional resources are distributed across different platforms.
5. The system of claim 1 wherein said non-transactional resources are resources of an Internet Data Center (IDC).
6. The system of claim 1 wherein said resource manager provides a proxy implementing a transactional protocol for said non-transaction resources.
7. The system of claim 6 wherein said transactional protocol is X/Open XA protocol.
8. The system of claim 1 wherein said at least one component is a plugin.
9. The system of claim 1 wherein said resource manager is communicatively coupled to a message bus.



10. The system of claim 9 wherein said message bus is an EAI bus.
11. The system of claim 1 wherein said resource manager is multi-threaded.
12. The system of claim 1 wherein said resource manager represents said transaction as an object.
13. The system of claim 12 wherein said resource maintains a log of the state of said object.
14. The system of claim 1 wherein said resource manager is operable in a plurality of different operational modes, which are definable by said at least one component.
15. A method of performing a functional service as a transaction utilizing one or more non-transactional resources within a computing environment, said method comprising the steps of:
  - at least one component defining one or more tasks executable by at least one of said one or more non-transactional resources;
  - receiving at a resource manager a request for performance of a plurality of tasks as a transaction; and
  - said resource manager controlling execution of said at least one component to perform said plurality of tasks as a transaction for service provisioning.
16. The method of claim 15 further comprising the step of:
  - client application requesting said functional service.
17. The method of claim 16 further comprising the step of:
  - message bus communicatively coupled to said client application receiving said request for said functional service and redirecting said request to one or more proper resource adapters.
18. The method of claim 17 wherein said resource manager acts as an intermediary between said one or more resource adapters and said one or more non-transactional resources to control said non-transactional resources to perform said plurality of tasks as a transaction.

19. The method of claim 18 wherein said resource manager interacts with a transaction manager via transactional protocol.

20. The method of claim 18 further comprising:  
said resource manager invoking tasks at said at least one component according to a transactional protocol.

21. The method of claim 20 wherein said transactional protocol is X/Open XA protocol.

22. The method of claim 15 wherein said at least one component is a plugin component.

23. A resource manager operable to control execution of tasks by one or more non-transactional resources to perform said tasks as a transaction, said resource manager comprising:

code for receiving a request for performance of a plurality of tasks;  
code for controlling one or more non-transactional resources to perform said plurality of tasks as a transaction, wherein said code for controlling one or more non-transactional resources includes code for invoking performance of a task by said one or more non-transactional resources, and wherein said code for invoking performance of a task includes code for calling a function defined by a plugin component that is communicatively coupled to said one or more non-transactional resources.

24. The resource manager of claim 23 further comprising:  
code for representing said transaction as an object.

25. (Canceled)

26. (Canceled)

27. A system comprising:  
non-transactional resources for providing a plurality of different services;  
a plugin for each of said plurality of different services that defines one or more tasks for activating the respective service; and  
resource manager operable to control execution of said plugins to selectively activate multiple ones of the plurality of different services as a transaction.

28. The system of claim 27 wherein said plurality of different services comprise ones selected from the group consisting of:  
web hosting service, database service, software application service, DNS service, LDAP service, monitoring service, management service, quality of service (QoS) service, usage measurement service, and billing service.

29. A system comprising:  
at least one non-transactional resource for providing at least one service associated with a web hosting service;  
a plugin for said at least one non-transactional resource that defines one or more tasks for activating the at least one associated service for a given web hosting service; and  
resource manager operable, when activating said given web hosting service, to control execution of said plugin to activate said at least one associated service as a transaction with activating said given web hosting service.

30. The system of claim 29 wherein said at least one associated service includes a plurality of different services associated with a web hosting service.

31. The system of claim 30 wherein the plurality of different associated services comprise ones selected from the group consisting of:  
database service, software application service, DNS service, LDAP service, monitoring service, management service, quality of service (QoS) service, usage measurement service, and billing service.